A Neural Network Estimation of Ankle Torques From Electromyography and Accelerometry

Ho Chit Siu, Jennifer Sloboda, Ryan J. McKindles, and Leia A. Stirling, *Member, IEEE*

***Abstract*— Estimations of human joint torques can provide clinically valuable information to inform patient care, plan therapy, and assess the design of wearable robotic devices. Predicting joint torques into the future can also be useful for anticipatory robot control design. In this work, we present a method of mapping joint torque estimates and sequences of torque predictions from motion capture and ground reaction forces to wearable sensor data using several modern types of neural networks. We use dense feedforward, convolutional, neural ordinary differential equation, and long short-term memory neural networks to learn the mapping for ankle plantarflexion and dorsiflexion torque during standing, walking, running, and sprinting, and consider both single-point torque estimation, as well as the prediction of a sequence of future torques. Our results show that long short-term memory neural networks, which consider incoming data sequentially, outperform dense feedforward, neural ordinary differential equation networks, and convolutional neural networks. Predictions of future ankle torques up to 0.4 s ahead also showed strong positive correlations with the actual torques. The proposed method relies on learning from a motion capture dataset, but once the model is built, the method uses wearable sensors that enable torque estimation without the motion capture data.**

***Index Terms*— Accelerometers, biomechanics, electromyography, neural networks, wearable sensors.**

I. Introduction

THE estimation of human joint torques is a common metric in the assessment of human biomechanics, which provides quantitative and clinically valuable information [1]. These estimates can be useful as part of a functional assessment, as well as for evaluating and planning patient care and therapy. Moreover, the estimation of joint torque for an individual person can support the design of wearable robotic systems, such as assistive exoskeletons. Though several exoskeleton technologies exist today, significant work remains to accurately predict human movement with wearable devices and optimize the actuation response [2]. Towards this goal, a method of predicting non-assisted human joint torques across relevant activities using wearable sensors can inform requirements for controller design.

Current approaches of estimating joint torques typically fall into two categories: a complete and a simplified inverse dynamics method [1]. In the first method, marker trajectories from motion capture systems are used to track human kinematics and ground reaction forces from force-sensitive platforms are used to track interaction with the external environment. These data are then combined with estimates of body segment inertia to estimate forces and moments using kinematic chain models. The second method still requires estimates of joint kinematics and ground reaction forces, but simplifies the problem by disregarding the inertial properties of the body segments and assumes a point mass for the person. Both of these methods typically incur substantial equipment and personnel costs and confine estimations to a laboratory setting. One alternative is to use a wearable exoskeleton specifically for measurement, as in Li *et al.* [3]. This method measures interaction forces between the human and the exoskeleton, and the combined human/exoskeleton kinematics directly to perform inverse dynamics. It removes the need for a motion capture setup, but requires the subject to bear the additional load of the exoskeleton itself as well as coupling the subject to a mechanical system that may affect their normal motions.

These methods for estimating inverse joint dynamics use models of the human body (or body segments) to estimate joint torques. Machine learning methods provide a different approach to torque estimation, which forgo the need for a human body model and instead learn a mapping directly from sensor input to torque or joint angle output, as shown by Jacobs and Ferris [4], and Dorschky *et al.* [5], respectively, both for walking. With machine learning, joint torque could be estimated using lightweight wearable sensors such as surface electromyography (sEMG) and inertial measurement units (IMU), among other input modalities, eliminating the need for expensive equipment or external hardware structures [6]. Moreover, this method enables real-time joint torque estimation outside of the confines of a laboratory or clinic after an initial data collection period, broadening the range of possibilities for dynamic assessments.

Our previous work in Siu *et al.* [6] evaluated the performance of feedforward and recurrent neural networks for ankle torque regression on a dataset comprised of standing, walking,

Manuscript received January 22, 2021; revised July 15, 2021; accepted August 5, 2021. Date of publication August 13, 2021; date of current version August 26, 2021. DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force. *(Corresponding author: Ho Chit Siu.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the MIT Committee on the Use of Humans as Experimental Subjects under Protocol No. 1703875483.

Ho Chit Siu, Jennifer Sloboda, and Ryan J. McKindles are with Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02421 USA (e-mail: hochit.siu@ll.mit.edu; jennifer.sloboda@ll.mit.edu; ryan.mckindles@ll.mit.edu).

Leia A. Stirling is with the Department of Industrial and Operations Engineering, Robotics Institute, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: leias@umich.edu).

Digital Object Identifier 10.1109/TNSRE.2021.3104761

running, and sprinting. We found that when using accelerometry and surface electromyography (sEMG) inputs from wearable sensors, neural networks that took the sequential nature of the data into account — recurrent neural networks (RNNs) and the more specialized RNN variant, long short-term memory (LSTM) networks — outperformed typical feedforward architectures from the literature in estimating instantaneous left and right ankle torques.

This study uses largely the same dataset as in [6], consisting of the aforementioned wearable sensors, and ground truth estimates of joint torque calculated from inverse dynamics that used full motion capture and a force-sensitive platform along with the Plug-in Gait Model (Vicon, Oxford, UK). We expand on the previous work in several ways. First, we compare the best-performing architectures from the previous work against two additional types of network architectures: convolution neural networks (CNNs) [5], [7] and neural ordinary differential equations [8]. Most of the previously-examined architectures, which were from the literature, were simple feedforward methods, and the ones considered here (in addition to the LSTM) represent more modern architectures. Second, we examine the ability to estimate not only instantaneous ankle torques, but also entire sequences of ankle torques, both for the same time period as the input data, as well as for predictions of torques after the time of the input data. Finally, the effects of data augmentation via oversampling are considered as a way to increase the training utility of small biomechanical datasets.

An interesting architecture that we examine here for biomechanics is the neural ordinary differential equation (NODE) [8]. This method relies on the characteristic that stacked feedforward layers in a typical feedforward network act as universal function approximators, and that the derivative of such a function can be solved in a continuous manner using numerical methods. NODE removes the discrete nature of individual network layers, and rather treats the entire network as having “continuous” depth, where evaluations of the value of the network at any hidden state can be found via a numerical ODE solver. Critically, it removes the need to set a network depth, as numerical solvers take on that role in an adaptive manner, effectively creating a “deeper” network only when the complexity of the function being approximated requires it.

The major contributions in this paper are threefold: (1) a comparison of the performance of several modern types of neural network models on the problem of ankle torque regression from wearable sensors, (2) the evaluation of sequence regressions for estimating a time series of torques, with and without data augmentation, and (3) the use of these methods to predict future torque sequences.

II. Literature Review

A. Human Motion and Joint Force Estimation From Wearable Sensors

Common approaches to estimating human motion and joint forces from wearable sensors typically rely on inertial measurement units and less often sEMG in clinical or laboratory settings. The use of IMUs to estimate human movement has gained popularity in the recent decades due to increasing ubiquity of “smart” devices (phones, watches, etc.), which typically contain such sensors and are carried close to the body [9]. Algorithms designed for these mobile devices typically perform human activity recognition, rather than finer-grained motion analysis, and model training is almost exclusively performed offline before deployment into a consumer or research product. Hidden Markov models [10], boosting [10], support vector machines [11], [12], and conditional random fields [13] are among the methods used for IMU-based activity recognition. Although not as common as IMUs, sEMG sensors are another modality used to classify and predict human motion [14]–[17]. sEMG sensors are placed on the skin to record electrical activity of underlying muscles, though the placement and selection of these body-worn sensors can have a significant impact on the accuracy of activity classification [18]. Information from sEMG can be used to estimate future movement trajectories and the associated joint forces and can serve as an anticipatory control signal for wearable devices [7], [19], [20]. sEMG signals are more sensitive than kinematic measurements to factors such as subject-specific physiology, placement, and noise, though machine-learned models have shown efficacy in motion classification tasks despite noise and nonspecific sensor placement [15], [17], [20], and locomotion classification using a combination of sEMG and accelerometers have been shown higher accuracy than with accelerometers alone [18].

B. Neural Networks for Biomechanical Parameter Estimation From Wearable Sensors

Neural networks are a popular machine learning technique that have shown utility for a number of applications in biomechanics. A very common type of neural network is a simple fully-connected (or *dense*) feedforward network architecture. In this architectures, data moves unidirectionally through a fixed set of layers, where each unit in one layer is fully connected to every unit in the next layer through a set of weights and biases. Such networks have been used previously to estimate ground reaction forces and ankle moments during walking and calf raises [4], ankle angles and moments in walking [21], and energy expenditure during locomotion [22].

More sophisticated types of neural networks have been developed to capture temporal or sequential information from the input data. Recurrent neural networks (RNNs) give the previous output as an input back into the network for predicting the next time step. This structure makes RNNs advantageous for sequential data, but also introduces difficulties in training. Song and Tong [23] used an RNN to estimate elbow torque from sEMG sensors in a dynamic tabletop manipulandum task. However, the standard RNN that they used suffers from the vanishing gradient problem during training, *i.e.*, long sequences of inputs are not learned well because the information carried by the gradient during backpropagation diminishes the longer the length of the input sequence [24].

Long short term memory networks (LSTMs) are a kind of RNN that attempt to avoid the vanishing gradient problem [24], [25] by using a recurrent structure that regulates the flow of previously-seen data by encoding a *cell state* variable that is modified based on current inputs and previous cell states before being passed along. The training of these networks tunes the degree to which the existing cell state and new information is used to modify the cell state, which ultimately determines the network output. The use of cell states rather than a simple recurrent unit allows information to be "gated" and prevents the vanishing gradient problem. The rest of this paper uses the abbreviation RNN to refer to standard recurrent neural networks, and LSTM will refer to this specific variant.

Much like "deep" variants of dense networks, a stacked LSTM architecture allows more complex representations of the data to be developed than would be possible with a single LSTM layer [26]. However, the data requirements of these larger networks grow substantially as the number of tunable parameters increases. A two-layer LSTM was implemented by Slade *et al.* [22] for estimating energy expenditure during locomotion, but was found to perform worse than a dense network. However, they reported computational difficulties that limited their use of an LSTM to a subset of the data that was used for the dense feedforward network, which may have contributed to the lower accuracy.

Convolutional neural networks (CNNs) are another type of network, characterized by layers that consider the spatial relations of incoming data through the use of (typically 2D) convolutions. CNNs have become *de facto* parts of image-based machine learning [27], but have also been used for mapping wearable sensor data to biomechanical parameters [5], sometimes in combination with LSTMs [7]. In biomechanics applications, CNNs may be structured to consider segments of data in their temporal context directly, without the additional use of RNNs/LSTMs, but in such cases, they do not have the strict sequential processing of the latter two architectures. However, they have a notable speed advantage in some situations, since modern graphics processing units (GPUs) are optimized for parallel processing of the many matrix multiplications on which CNNs rely.

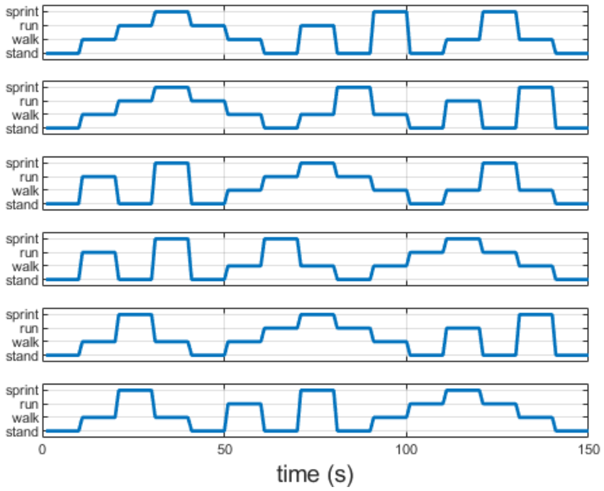

Fig. 1. Sequence of locomotion commands given to subjects across the six locomotion trials. Each plot represents a trial. Since subjects were on a self-paced treadmill, their locomotion may have varied from the commands given.

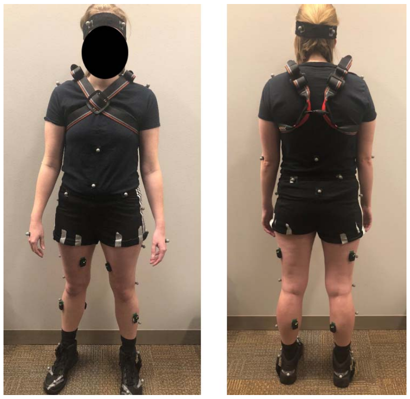

Fig. 2. Example sensor and marker setups for one subject with wireless accelerometer/sEMG sensors and motion capture markers. Subjects wore a chest harness as part of a safety system for the treadmill.

III. Methods

A. Data Collection and Preprocessing

Five subjects performed the experiment protocol, but due to sensor malfunction, only four subjects' data were usable. From these four subjects, there were three female and one male, ages 27, 21, 31, and 21. All subjects were asked to walk on a self-paced, split-belt treadmill. Each subject completed six trials of 150 seconds in duration that were presented in the same order for each subject. For each trial, subjects were given visual commands to "stand," "walk," "run," and "sprint," for 10-second blocks (Figure 1), with breaks between trials (variations in subject speeds are shown in Table II). In total, each subject spent 300 seconds standing, 240 seconds walking, 180 seconds running, and 180 seconds sprinting. All subjects provided written, informed consent and the protocol was approved by the MIT Committee on the Use of Humans as Experimental Subjects (protocol #1703875483). This is the same underlying dataset as in Siu *et al.* [6], with one additional subject.

For all trials, subjects wore eight wireless sEMG sensors with embedded three-axis accelerometers (Delsys, Natick, MA) with four sensors on each leg (tibialis anterior, medial gastrocnemius, vastus medialis, and semitendinosus), where electrodes were positioned on top of the center of the muscle bodies after palpation, and parallel to the muscle. These muscles were chosen as they are primary contributors to ankle and knee motion. Motion capture markers were placed on the subjects in a modified Plug-In Gait model configuration with the full lower-body marker set and a reduced upper-body marker (Vicon, Oxford, UK) (Figure 2). Throughout the experiment, force plates embedded under each tread (left and right) of the treadmill were sampled at 1000 Hz, sEMG and accelerometers were sampled at 2000 Hz, and the motion capture data were sampled at 100 Hz. Motion capture and ground reaction force data were combined in the Plug-in Gait model to calculate the inverse dynamics-derived joint torques, which were the labels for the machine learning training. All collected signals were downsampled to 100 Hz for further processing, a sample rate that was chosen to be similar to what would be feasible for

TABLE I
REGRESSION METHODS

Method	**Parameter Count**	**Architecture Detals**	**Learning Rate**
Dense Feedforward	30642	6 dense layers with 80 hidden units per layer	0.0005
CNN	156264	1 convolution-ReLU layer with filter size 8, kernel size 5x3; 1 max-pooling layer with pool size 2x2; 1 convolution-ReLU layer with filter size 16, kernel size 5x3; 1 max-pooling layer with pool size 2x2; 1 flattening layer with 1440 hidden units; 1 dense-ReLU followed by 1 dense layer with 100 hidden units each	0.0001
Neural ODE	30912	continuous-depth with 110 hidden units	0.0001
LSTM	32306	1 LSTM layer with 64 hidden units and 2 dense layers with 16 hidden units per layer	0.001

Model parameters were chosen such that all models had $\approx 30,000$ tunable parameters, except the CNN, which was based on [5]. All models used leaky ReLU activations except for the final output layer, which was linear.

TABLE II
SPEED VARIATION IN SELF-PACED TREADMILL LOCOMOTION

Instruction	Subject 1	Subject 2	Subject 3	Subject 4
stand	$0.0, 0.0, 0.0$	$0.0, 0.0, 0.0$	$0.0, 0.0, 0.0$	$0.0, 0.0, 0.0$
walk	$1.8, 1.1, 1.3$	$1.9, 1.2, 1.4$	$0.7, 1.0, 1.2$	$0.9, 1.2, 1.5$
run	$1.9, 2.6, 3.0$	$2.2, 2.6, 3.0$	$1.7, 2.1, 2.6$	$2.0, 2.5, 2.8$
sprint	$2.0, 3.6, 4.4$	$2.1, 3.6, 4.2$	$2.7, 3.8, 4.5$	$2.4, 3.2, 3.7$

25th, 50th, and 75th percentile speeds (m/s) for all subjects. Quantiles shown instead of mean and standard deviation to avoid outlier skewing (e.g. during transitions).

use with an online motion prediction system using the feature generation and inference methods described here.

B. Feature Extraction

Downsampled signals were processed to extract features, without additional filtering or rectification, at the same 100 Hz rate, using a 0.5 s historical window of sensor signals, and 0.49 s overlap with adjacent windows, similar to [15]. sEMG features were the max value and area for each window. Accelerometer features were the median vector magnitude and the median angle in the X-Y, Y-Z, and Z-X planes for each window. In the case of Subject 4, the integrated sensors also included a gyroscope, for which the mean magnitude of the angular velocity was also calculated for each window. We decided to include the gyroscope features, which were only available for Subject 4, to give the models as much data to work with as possible as the comparisons being made in this study are focused on the model architectures rather than sensor inputs. These features were selected due to their relatively low computational cost, making them well-suited to wearable device applications. The features were subsequently used as the inputs to the machine learning models.

C. Ankle Torque Regression Across Neural Network Architectures

To evaluate regressions across model architectures and provide updated comparisons from our previous work in [6], we consider four classes of neural networks: a dense (fully-connected) feedforward network, a convolutional network, a neural ordinary differential equation network, and long short-term memory network. The chosen architectures are summarized in Table I. Each architecture was trained using the same inverse dynamic model output from the Nexus Plug-In Gait model. For all but the CNN model, we controlled the number of tunable parameters to be approximately 30,000 (when outputting a single pair of torques) and used leaky ReLU activations and Adam optimization for consistency. Since the neural ODE requires equal-sized inputs and outputs for the continuous-depth section of the network, linear reshaping layers were added before and after to ensure size consistency. The CNN could not use a similar number of parameters due to the requirements for parameters connecting convolution layers, so instead, we replicated a network similar to the one used in [5]. Learning rate sweeps were performed in log-linear steps from 10^{-7} to 10^{-3} to find the best learning rate for each model. No learning rate decay was used.

A subject-specific six-fold cross validation was used with five training trials and one test trial, where these trials correspond to the experiment trials in Figure 1. *Sequences* of features from fifty contiguous time steps were used as model inputs. For each of the models, the instantaneous left and right ankle plantarflexion/dorsiflexion torques at the end of the data sequence were the two regression targets, and each model output a regression of both ankle torques at once. We will refer to this as the *sequence-to-one* regression paradigm.

For the regression outputs, both the root mean square error (RMSE) and the Pearson correlation (ρ) between the estimated and inverse dynamics torque values were calculated as measures of accuracy. The accuracy results are reported for each model for both the pooled and individual locomotion activities, even though models were not trained independently for each activity. Each ankle is considered an independent subgroup for the purposes of calculating the Pearson correlation.

D. Sequence-to-Sequence Ankle Torque Regression

In addition to comparing regression performance for instantaneous torque across different model architectures, we also consider the problem of *sequence-to-sequence regression* for the LSTM architecture. In these experiments, the same training and testing procedure is followed, but instead of estimating a single pair of left and right foot target values (torques) per set of feature inputs, models are made to predict torque values for each time point in the provided sequence, resulting in $2 \cdot T$ regression outputs for each $f \cdot T$ input values, where T is

TABLE III
REGRESSION MEAN SQUARED ERROR (RMSE) BY MODEL AND ACTIVITY

	All Activities	Stand	Walk	Run	Sprint
Dense Feedforward	0.10 ± 0.06	0.04 ± 0.03	0.08 ± 0.03	0.11 ± 0.06	0.15 ± 0.12
CNN	0.10 ± 0.05	0.04 ± 0.03	0.09 ± 0.07	0.11 ± 0.03	0.14 ± 0.06
Neural ODE	0.11 ± 0.12	0.06 ± 0.08	0.11 ± 0.17	0.11 ± 0.06	0.15 ± 0.14
LSTM	$\mathbf{0.08 \pm 0.02}$	$\mathbf{0.04 \pm 0.01}$	$\mathbf{0.07 \pm 0.02}$	$\mathbf{0.09 \pm 0.03}$	$\mathbf{0.11 \pm 0.03}$

Means and standard deviations of root mean square error ($N \cdot m/kg$) over cross-validation folds. Results are pooled from all subjects. See Table I for model definitions.
One outlier fold from dense feedforward from Subject 1 was rejected due to failure to converge. Otherwise, all values represent 24 folds total (6 folds, 4 subjects).

TABLE IV
REGRESSION PEARSON CORRELATION BY MODEL AND ACTIVITY

	All Activities	Stand	Walk	Run	Sprint
Dense Feedforward	0.85 ± 0.04	0.72 ± 0.08	0.81 ± 0.08	0.89 ± 0.05	0.87 ± 0.04
CNN	0.80 ± 0.05	0.62 ± 0.09	0.71 ± 0.06	0.86 ± 0.05	0.82 ± 0.05
Neural ODE	0.85 ± 0.05	0.73 ± 0.09	0.81 ± 0.05	0.88 ± 0.05	0.88 ± 0.05
LSTM	$\mathbf{0.88 \pm 0.03}$	$\mathbf{0.76 \pm 0.07}$	$\mathbf{0.84 \pm 0.06}$	$\mathbf{0.91 \pm 0.04}$	$\mathbf{0.90 \pm 0.04}$

Means and standard deviations over the six-fold cross-validation, with each joint treated as a separate subgroup. Results are pooled from all subjects. See Table I for model definitions.

the length of both the feature and output sequences and f is the number of input features per time point ($f = 112$). For the sequence regressions, we use $T = \{50, 150, 300\}$, corresponding to 0.5, 1.5, and 3.0 s of data. 0.5 and 1.5 s lengths are slightly less than and slightly greater than a typical walking gait cycle, and 3.0 s was used as a stress-test, as it provided a much longer input/output sequence.

E. Data Augmentation Comparisons

We consider the effects of training data augmentation for the LSTM architecture. We compare the performance of models trained with unaugmented data against those trained with data augmented via oversampling in three ways: by activity class (*activity augmentation*), by maximum torque value in a sequence (*max augmentation*), and by the range of the torque values in a sequence (*range augmentation*). Oversampling is a commonly-used technique for improving performance of deep learning models on under-represented data [28]. These approaches were chosen in an effort to improve predicted peak/trough amplitude accuracy in run and sprint activities, as this was an observed shortcoming in unaugmented models.

For activity augmentation, under-represented classes in the training set are randomly oversampled until the number of instances approximately matches that of the most represented class. The on-screen command given to the subject is taken as a proxy for the true activity class label. Since running and sprinting have similar biomechanics, we consider them a single class for activity oversampling.

When augmenting by the other two methods, we calculate a *characteristic value* for each target sequence: the maximum torque value (for max-augmentation), or the range between the minimum and maximum torque values (for range augmentation). The characteristic values are then grouped into five histogram bins, and oversampling is performed as in the case of activity-based augmentation, treating each bin as a class. The results of data augmentation are compared against models trained under the same conditions, but without augmentation.

F. Future Torque Predictions

Finally, we apply the LSTM model to the task of predicting a sequence of future torques, beyond the time period of the provided sensor inputs. Since we are associating future torque labels with past sensor data as part of future prediction, we consider first whether a “burn-in” time is required for the LSTM to reach steady-state error. This occurs because LSTMs produce outputs sequentially, and the first outputs are produced when the LSTM memory is still being initialized. While burn-in would not affect our augmentation and sequence length comparisons (since they would consistently occur), we measure it explicitly for these comparisons in order to remove outputs from the burn-in period. All outputs after the burn-in are for times that are after the time stamp of the last sensor reading in the input, thus giving a torque prediction for the immediate future.

IV. RESULTS

A. Neural Network Architectures

Two model accuracy metrics are reported for the sequence-to-one regressions in Tables III and IV as root mean square error and Pearson correlation, respectively. The LSTM model had the lowest RMSE on average and across all activities (Table III). Conversely, the dense feedforward and NODE had higher error in many cases, even after the exclusion of two outlier folds that existed for Subject 1 (one that did not converge, and one that was simply higher error). The non-converging fold for dense feedforward was also a particularly high-error outlier for NODE throughout all parameter sweeps for both, indicating that it may have been due to the data from that fold, rather than the learning method that caused the problem. Though the same fold resulted in higher error for the LSTM, it did not cause the numerical instability that occurred with the feedforward model. Across all models, sprinting resulted in the highest average error across activities, due to the larger torques involved and errors in estimating very brief peak torques.

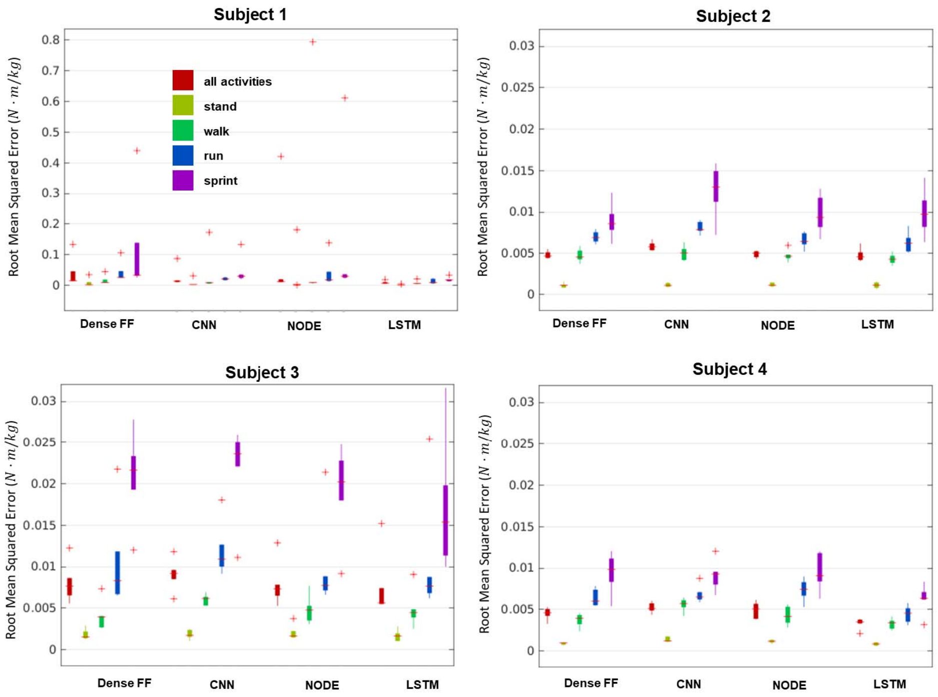

Fig. 3. Root mean square error from all models by activity and subject. The same fold for Subject 1 that was excluded in Table III was also excluded here. Note that Subject 1 has different y axis limits than the rest.

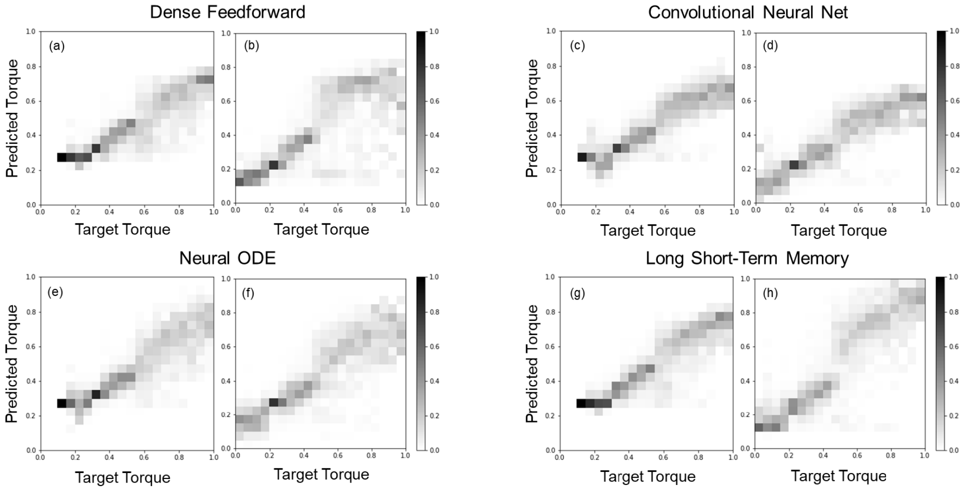

Fig. 4. 0-1 normalized ankle torque estimates (rows) vs target ankle torques (columns) using four methods for one fold from Subject 3. Each pixel column has been also been normalized to sum to one in order to highlight cases with higher target torques, which occur much less frequently than lower torques. Perfect agreement would mean a single line along the diagonal.

Similarly, the LSTM models resulted in the highest overall correlation values with the ground truth joint torque values (Table IV) across all activities.

Inter-subject variability was observed across model classes and activity conditions (Figure 3). Apart from outliers, all subjects had a similar order of magnitude of errors across models, with a persistent pattern of higher error during higher-torque locomotion.

For a representative subject and validation fold, heat maps of the torque estimates vs target values are presented (Figure 4). In these plots, a perfect set of regressions would appear as a dark stripe along the diagonal. Here, we see that most the dense feedforward and CNN tended to underestimate higher target torque values, and a general trend of greater variance in higher target torques can also be observed.

B. Sequence-to-Sequence Regression and Data Augmentation

A summary of the RMSE values from an LSTM architecture on varying target lengths for sequence-to-sequence regression with the three types of data augmentation is shown in Figure 5. The augmentation methods showed mixed results compared to unaugmented training. Augmentation reduced the interquartile

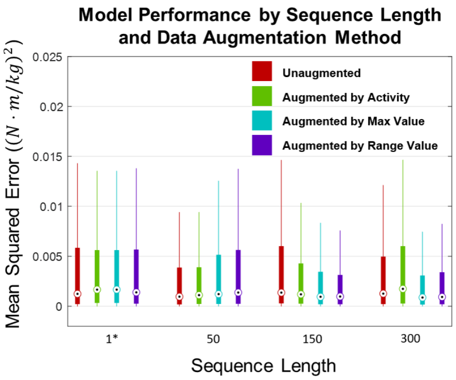

Fig. 5. Mean square errors from different target sequence lengths (LSTM). All results use equal-length feature and target sequences (sequence-to-sequence regression), except for 1*, which uses a length-50 feature sequence and a length-1 target (sequence-to-one regression). For clarity, outliers are not shown. In all cases, all outliers were on the upper end of the distribution, and represented 11%-13% of the corresponding sample.

range in the longer sequences (150, 300), particularly with max and range augmentation. Distributions in sequence-to-one regression remained the same, and augmentation may have a negative effect in 0.5 s torque regression (Figure 5 and Figure 6, a and c). Although max-value augmentation improved the match between the high-torque periods of run (Figure 6, b vs. d), it degrades estimation during walk (a vs. c), flattening the prediction (c) compared to the unaugmented prediction (a). 3 s regressions showed similar differences between unaugmented and max-augmented run gait cycles (Figure 6, b vs. d and f vs. h), but walk gait cycles regression (a vs. c) were generally flatter and closer to the cycle's mean value than in the 0.5 s case, both with and without augmentation.

C. Future Torque Prediction

Our burn-in analysis (Figure 7) showed that for a length-50 (0.5 s) input and output sequence, a transient with consistently high error occurred in the first 5-10 outputs produced by the LSTM. Thus, for all subsequent future prediction analysis, we shifted the torque labels such that there was 0.1 s overlap with the end of the sensor data, and 0.4 s beyond the sensor data. The first 0.1 s of each prediction was then removed. The results for this future prediction, using activity-augmented data is shown in Tables V and VI. These LSTM sequence predictions show similar RMSE error to single-point regression, but have higher variance (Tables III and V), and lower (though still positive) correlation. Best- and worst-case prediction outputs for walking and run/sprint are shown as gait cycle plots in Figure 8 as examples.

V. Discussion

A. Measurement Comparisons

In this work, we sought to broadly consider the effects of using different types of modern neural network architectures on a biomechanics mapping problem, and for the LSTM, consider the task of future torque prediction. In contrast to our previous comparisons of architectures used in the literature [6], results between architectures here were less varied, likely because the overall greater sophistication of these networks allowed better learning of the task. Overall, the LSTM models had higher average single-point estimation accuracy (lower RMSE and higher correlation) than other methods (Tables III and IV), even after outliers were rejected from Dense Feedforward and Neural ODE. Moreover, the ability of the LSTM to perform well even in cases that led to numerical instabilities in the feedforward model and additional outliers in both the feedforward and the NODE means that the LSTM may be a more stable method for these applications. Importantly, we also see that the CNN did not perform notably better than the other models (and indeed had the lowest overall correlation) despite the fivefold greater number of tunable parameters compared to the other models.

We found that a multi-metric approach was required to appropriately describe model accuracy for joint torque estimations during diverse movement activities. During quiet standing, the average error was low, but the Pearson's correlation was also low (Tables III and IV). This difference likely emerges from the relatively low amplitude joint torque required for standing resulting in small differences when compared to the inverse dynamics torques. Therefore, RMSE may not be a particularly good indicator of accuracy for low-torque activities. The walking gait cycles also illustrate this point, as the regressions are qualitatively worse than those for run/sprint (Figure 6). Conversely, high error occurred during sprinting (2-3 times as high as during stand), while Pearson's correlation (≈0.8-0.9) was consistent with the other activities. Finally, average accuracy values observed for a person or activity do not describe the torque level-specific changes. The heat maps and gait cycle plots provide an improved understanding during which parts of the gait cycle the models are providing the most accurate estimates of joint torque and where the algorithms could be improved.

Nuances of describing model accuracy are also highlighted in our future prediction results. Visual inspection by gait cycle (Figure 8) shows that for some subjects, peak torques are still potentially prone to underprediction, and peak timing may be mis-predicted; RMSE does not account for such differences directly. Gait parameters (e.g. peak timing and magnitude) may be used with pre-defined torque profiles in some applications, with differing emphasis placed on each, perhaps affecting the choice of loss function.

The data used here present a challenge that can be seen from the mixed results of our augmentation efforts. Augmentation only appeared to benefit the longer sequences that were 1.5 and 3.0 s in length. Run and sprint periods are *overrepresented* in terms of number of gait cycles (due to shorter gait cycles than when walking), but are *underrepresented* in terms of their torque characteristics, both in terms of the maximum values and the range of values contained in any sequence from these activities (again because of shorter cycles). Additionally, the higher torque values during run/sprint means that they contribute a disproportionately large amount to the RMSE. Activity-based augmentation resulted in similar error distributions to unaugmented training (Figure 5), but max and range

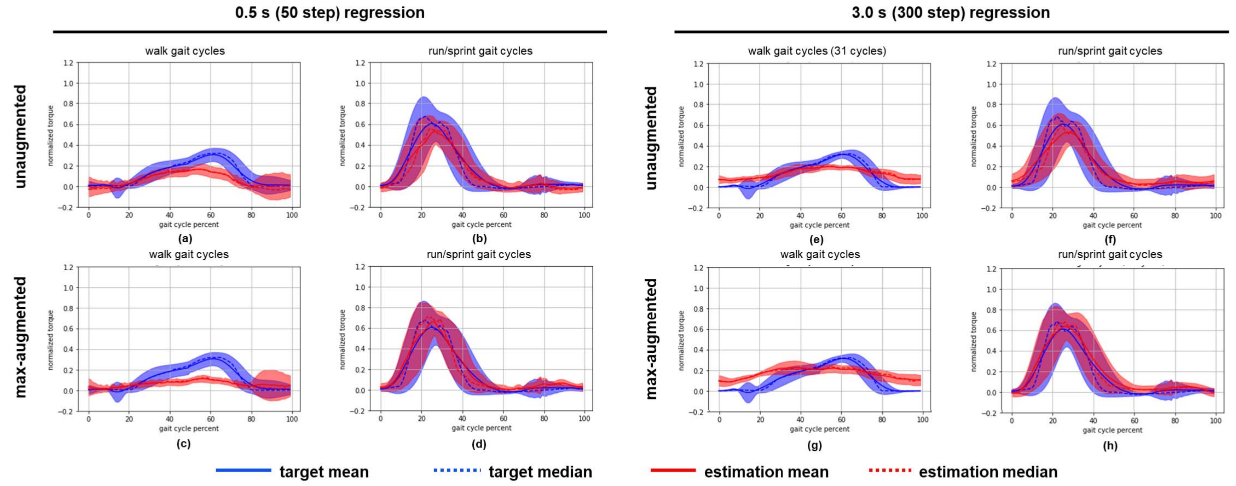

Fig. 6. Comparisons of two different output lengths (50 and 300, corresponding to 0.5 and 3.0 s) for sequence-to-sequence regression. Target (Plug-In Gait) LSTM-estimated gait cycle torques for one trial from Subject 3. Shaded regions are one standard deviation from the mean. Top and bottom rows of plots are unaugmented and max-augmented, respectively. Each plot represents 30-50 gait cycles.

TABLE V
LSTM FUTURE TORQUE PREDICTIONS ERRORS BY SUBJECT AND ACTIVITY

	All Activities	Stand	Walk	Run	Sprint
Subject 1	0.12 ± 0.21	0.06 ± 0.15	0.12 ± 0.22	0.14 ± 0.20	0.17 ± 0.25
Subject 2	0.09 ± 0.16	0.04 ± 0.08	0.09 ± 0.14	0.11 ± 0.17	0.13 ± 0.21
Subject 3	0.06 ± 0.11	0.03 ± 0.05	0.05 ± 0.07	0.07 ± 0.10	0.10 ± 0.15
Subject 4	0.09 ± 0.15	0.04 ± 0.08	0.08 ± 0.14	0.10 ± 0.15	0.12 ± 0.19

Means and standard deviations of root mean square error $(N \cdot m/kg)$ over cross-validation folds.

TABLE VI
LSTM FUTURE TORQUE PREDICTIONS CORRELATIONS BY SUBJECT AND ACTIVITY

	All Activities	Stand	Walk	Run	Sprint
Subject 1	0.54 ± 0.26	0.68 ± 0.39	0.51 ± 0.28	0.56 ± 0.12	0.51 ± 0.17
Subject 2	0.66 ± 0.22	0.44 ± 0.25	0.40 ± 0.27	0.75 ± 0.29	0.73 ± 0.17
Subject 3	0.73 ± 0.26	0.72 ± 0.43	0.71 ± 0.32	0.73 ± 0.26	0.73 ± 0.02
Subject 4	0.76 ± 0.29	0.43 ± 0.31	0.59 ± 0.33	0.82 ± 0.29	0.81 ± 0.06

Means and standard deviations of correlation over cross-validation folds.

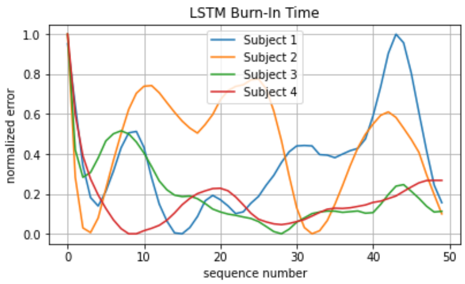

Fig. 7. Min-max normalized mean error as a function of the sequence number of an LSTM future prediction. Note the consistently higher error present earlier in the prediction sequences. From this, a burn-in time of 10 steps (0.1 s) was chosen for future prediction analysis.

data augmentation methods reduced overall error in some cases (Figure 5), primarily benefiting run/sprint estimates at the expense of walk estimates. Other ways to characterize the locomotion data for oversampling or non-oversampling augmentation methods could be explored, in addition to using more task-specific network architectures (e.g. with subnetworks for different activities). Related issues with data oversampling and other types of data augmentation are commonly found in machine learning and data mining literature, with the bulk of existing work addressing classification problems (i.e., image classification, fraud detection, text topic labeling, medical diagnosis) but a substantial and growing base addressing regression tasks [29], [30]. Such imbalance may be addressed in data-preprocessing (e.g., oversampling), algorithm design (e.g., modified neural network cost functions), post processing, or a hybrid of the aforementioned stages; there is no one-size-fits-all solution. In general, the data characteristics emphasized through data augmentation approaches ought to match the intended application, for example, peak amplitude and timing for ankle torque assisting exoskeletons. It should be noted that models trained specifically for a single locomotion type would not encounter the same challenges we experienced with under- or over-representation of gait cycles and target value ranges, but would have narrower application.

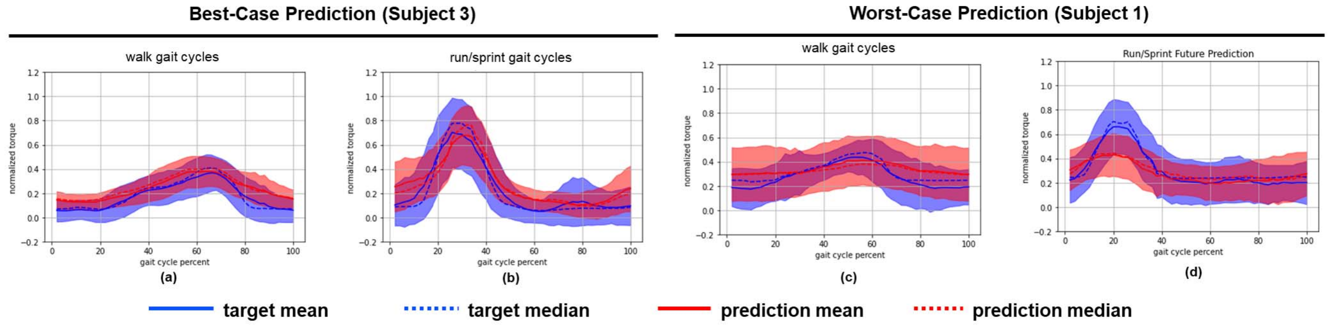

Fig. 8. Best- and worst-case LSTM predictions of future torques up to 0.4 s in advance, by gait cycle percentage for walk and run/sprint.

B. Implications for Measurement and Robotic Control

Current modeling methods are limited by the accuracy of the labels provided during training, and critically, by the volume of data. It may be possible with a larger dataset for neural networks to provide more accurate estimates despite inaccuracies in the labels (*e.g.* Rolnik *et al.*'s work on classifying images with corrupted training datasets [31]). However, neural network robustness to label noise is less explored for regression than for classification and will require further work. Additionally, it may be possible to learn appropriate estimations across individuals (testing on an unseen individual's data), as was done to some degree of success by Slade *et al.* for metabolic expenditure measurements [22].

Apart from our work, Jacobs and Ferris [4] also showed an example of ankle torque estimation during walking using a neural network and wearable sensors. Our results are not directly comparable due to sensor and procedure differences, but we would expect that improvements seen here in using an LSTM would also be transferable to their wearable system.

The ability to predict future joint torques may find useful applications to robotic control, as a form of human intent prediction. However, the best way to use such predictions (e.g. direct torque prediction or higher-level parameters) remains to be seen. Additionally for the case of wearable robotics, any feedback effects on prediction capability when the human is affected by external actuation remains to be tested.

C. Limitations

The first major limitation of this study is the small number of subjects ($N = 4$) and the subject-specific models. While the results of these experiments are promising on an individual level, variability in physiology, sensor placement, and other factors affect the generalizability of these results. However, the consistency of the accuracy trends between activities and between neural network architectures is promising (Figure 3).

Our activity labels were based on commands given to subjects, but subjects were on a self-paced treadmill. They were not restricted to a specific gait speed and were able to interpret the instructions differently (Table II). Similarly, transition periods between different activity conditions (*e.g.,* stand to walk) were also included in the analysis and may have lowered the accuracy reported for the steady state conditions. In future studies specific transition states may be uniquely identified and treated separately.

All our models output left and right ankle torques simultaneously. This architecture has reduced computational cost compared to independent left/right networks, but also means that each ankle torque estimation had access to sensor information from the opposite leg. While this may apply to some cases of torque estimation, we should note that we did not evaluate unilateral estimates independently.

Finally, the estimation models evaluated in this study all assumed signal stationarity. Though subjects were given breaks between trials, fatigue and task adaptation may alter the underlying neuromechanics of gait during experimental sessions. To compensate for these changes, models that have longer-term dependencies may be required over medium-term (*e.g.* a warehouse workday, a ruck march, etc.) and long-term (*e.g.* increased strength and conditioning over week or months, repetitive strain injuries, etc.) periods.

D. Future Work

A major barrier to operationalizing the work shown here is the requirement that subject-specific data be collected in a space with motion capture and instrumented force plates. Differences between human subjects means the creation of a large dataset of diverse biomechanical data remains a challenge, though this difficulty may be alleviated as consumer-grade wearable sensors become more powerful and less expensive.

Along similar lines, though we explored oversampling as a method of data augmentation, another type of augmentation — synthetic data generation — may also be useful. These methods use existing data to generate new, similar data, using approaches such as dynamic time warping [32] and generative adversarial networks [33]. Synthetic data generation has been shown to be effective for biosignal data augmentation, though most studies that specifically test these types of augmentation do so for classification, rather than regression problems. Additionally, lack of diversity in the underlying dataset — both of movements and participant physiology — is unlikely to be solved through augmentation alone.

Finally, the robustness of these methods to real-world conditions and external perturbation must be considered. In operational use cases, issues such as sensor dropout or irregularly-spaced data streams are common. If these predictions are used to control wearable robotics, the effects of external physical perturbation by the robot itself also needs to be examined.

VI. Conclusion

In this study, human ankle torques were estimated using sEMG and accelerometer data, leveraging subject-specific models trained on torques from motion-capture-based inverse dynamics. The effects of oversampling-based data augmentation on network training were also evaluated, to mixed results. A long short-term memory model provided the best performance, and gave similar performance when estimating instantaneous and sequences of torques up to 3 s long. The model also showed promise as a way to predict immediate future human torques, beyond the range of available data. The ability to use wearable sensors to obtain joint dynamics estimations and future predictions may be useful for both clinical and wearable robotic control applications.

Acknowledgment

The authors would like to thank Harvey Edwards and Aaron Rodriguez for their help in executing the experiments for this study, Paul Stegall for comments on the article, and Divya Chowbey and Emilia Grzesiak for improvements to the visualizations and machine learning code. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

[1] V. Camomilla, A. Cereatti, A. G. Cutti, S. Fantozzi, R. Stagni, and G. Vannozzi, "Methodological factors affecting joint moments estimation in clinical gait analysis: A systematic review," *Biomed. Eng. OnLine*, vol. 16, no. 1, p. 106, Dec. 2017.

[2] D. P. Ferris and B. R. Schlink, "Robotic devices to enhance human movement performance," *Kinesiol. Rev.*, vol. 6, no. 1, pp. 70–77, Feb. 2017.

[3] M. Li, J. Deng, F. Zha, S. Qiu, X. Wang, and F. Chen, "Towards online estimation of human joint muscular torque with a lower limb exoskeleton robot," *Appl. Sci.*, vol. 8, no. 9, p. 1610, Sep. 2018.

[4] D. A. Jacobs and D. P. Ferris, "Estimation of ground reaction forces and ankle moment with multiple, low-cost sensors," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, p. 90, 2015.

[5] E. Dorschky, M. Nitschke, C. F. Martindale, A. J. van den Bogert, A. D. Koelewijn, and B. M. Eskofier, "CNN-based estimation of sagittal plane walking and running biomechanics from measured and simulated inertial sensor data," *Frontiers Bioeng. Biotechnol.*, vol. 8, p. 604, Jun. 2020.

[6] H. C. Siu, J. Sloboda, R. J. McKindles, and L. A. Stirling, "Ankle torque estimation during locomotion from surface electromyography and accelerometry," in *Proc. 8th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Nov. 2020, pp. 80–87.

[7] L. Xu, X. Chen, S. Cao, X. Zhang, and X. Chen, "Feasibility study of advanced neural networks applied to sEMG-based force estimation," *Sensors*, vol. 18, no. 10, p. 3226, 2018.

[8] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," 2018, *arXiv:1806.07366*. [Online]. Available: http://arxiv.org/abs/1806.07366

[9] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, p. 33, 2014.

[10] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, 2005.

[11] T. Huynh, U. Blanke, and B. Schiele, "Scalable recognition of daily activities with wearable sensors," in *Proc. Int. Symp. Location-Context-Awareness*. Berlin, Germany: Springer, 2007, pp. 50–67.

[12] A. Bulling, J. A. Ward, and H. Gellersen, "Multimodal recognition of reading activity in transit using body-worn sensors," *ACM Trans. Appl. Perception*, vol. 9, no. 1, pp. 1–21, Mar. 2012.

[13] U. Blanke and B. Schiele, "Remember and transfer what you have learned-recognizing composite activities based on activity spotting," in *Proc. Int. Symp. Wearable Comput. (ISWC)*, Oct. 2010, pp. 1–8.

[14] F. Bai and C.-M. Chew, "Muscle force estimation with surface EMG during dynamic muscle contractions: A wavelet and ANN based approach," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 4589–4592.

[15] M. T. Wolf *et al.*, "Decoding static and dynamic arm and hand gestures from the JPL BioSleeve," in *Proc. IEEE Aerosp. Conf.*, Mar. 2013, pp. 1–9.

[16] M. M. Ardestani *et al.*, "Human lower extremity joint moment prediction: A wavelet neural network approach," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4422–4433, Jul. 2014.

[17] H. Siu, J. Shah, and L. Stirling, "Classification of anticipatory signals for grasp and release from surface electromyography," *Sensors*, vol. 16, no. 11, p. 1782, Oct. 2016.

[18] S. Gonzalez, P. Stegall, H. Edwards, L. Stirling, and H. C. Siu, "Ablation analysis to select wearable sensors for classifying standing, walking, and running," *Sensors*, vol. 21, no. 1, p. 194, Dec. 2020.

[19] C. J. De Luca, "The use of surface electromyography in biomechanics," *J. Appl. Biomech.*, vol. 13, no. 2, pp. 135–163, May 1997.

[20] H. C. Siu, A. M. Arenas, T. Sun, and L. A. Stirling, "Implementation of a surface electromyography-based upper extremity exoskeleton controller using learning from demonstration," *Sensors*, vol. 18, no. 2, p. 467, Feb. 2018.

[21] M. Mundt *et al.*, "Estimation of gait mechanics based on simulated and measured IMU data using an artificial neural network," *Frontiers Bioeng. Biotechnol.*, vol. 8, p. 41, Feb. 2020.

[22] P. Slade, R. Troutman, M. J. Kochenderfer, S. H. Collins, and S. L. Delp, "Rapid energy expenditure estimation for ankle assisted and inclined loaded walking," *J. Neuroeng. Rehabil.*, vol. 16, no. 1, p. 67, Dec. 2019.

[23] R. Song and K. Y. Tong, "Using recurrent artificial neural network model to estimate voluntary elbow torque in dynamic situations," *Med. Biol. Eng. Comput.*, vol. 43, no. 4, pp. 473–480, Jul. 2005.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[27] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.

[28] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[29] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modelling under imbalanced distributions," 2015, *arXiv:1505.01658*. [Online]. Available: https://arxiv.org/abs/1505.01658

[30] A. F. Hilario, S. G. López, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Honolulu, HI, USA: Springer, 2018.

[31] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," 2017, *arXiv:1705.10694*. [Online]. Available: http://arxiv.org/abs/1705.10694

[32] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Data augmentation using synthetic data for time series classification with deep residual networks," 2018, *arXiv:1808.02455*. [Online]. Available: http://arxiv.org/abs/1808.02455

[33] S. Haradal, H. Hayashi, and S. Uchida, "Biosignal data augmentation based on generative adversarial networks," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 368–371.